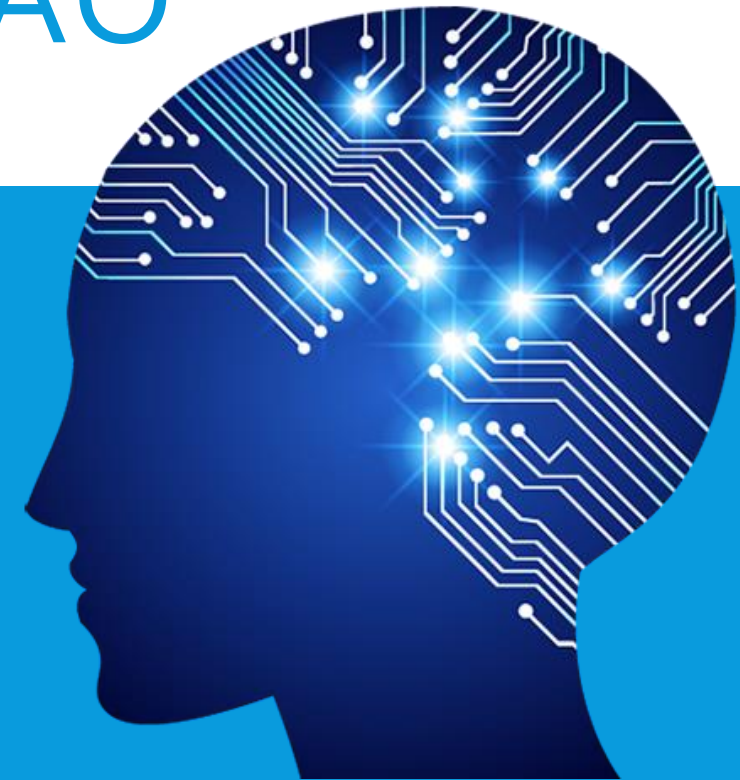


LÓGICA DE PROGRAMAÇÃO

Centro Universitário UniSenac – Campus Pelotas
Escola de Tecnologia da Informação
Prof. Edécio Fernando Iepsen



VETORES

- Permitem trabalhar com uma lista de dados
- Cada item da lista possui um índice

Tabela 6.1 – Representação dos itens/elementos de um vetor

produtos	
0	Arroz
1	Feijão
2	Iogurte
3	Leite
4	Suco
5	Pão

Para referenciar um item do vetor, devemos indicar seu nome, seguido por um número entre colchetes que aponta para o seu índice. É importante reforçar que o vetor inicia pelo índice 0. Portanto, para obter o primeiro produto inserido no vetor, devemos utilizar: `produtos[0]`.

EXEMPLOS

- Lista com o nome dos alunos de uma turma.
- Listas com descrição e preço dos produtos comprados em um supermercado.
- Listas de objetos, como os carros de uma revenda, onde cada objeto (carro) possui atributos: modelo, marca, ano e preço.

DECLARAÇÃO DE VETORES

```
const nomes = []
```

```
// ou
```

```
const nomes = new array()
```

DIFERENÇA ENTRE VARIÁVEIS E VETORES

- Uma variável armazena apenas um valor por vez; quando uma nova atribuição a essa variável é realizada, o seu valor anterior é perdido. Após as duas atribuições a seguir, a variável `idade` permanece apenas com o último valor que lhe foi atribuído.

```
let idade  
idade = 18  
idade = 15
```

- Já os vetores permitem armazenar um conjunto de dados e acessar todos os seus elementos pela referência ao índice que identifica cada um deles. Assim, após as duas atribuições a seguir, os dois valores atribuídos ao vetor `idade` podem ser acessados.

```
const idade = []  
idade[0] = 18  
idade[1] = 15
```

INCLUSÃO E EXCLUSÃO DE ITENS DE UM VETOR

Depois de realizarmos a declaração do vetor, podemos gerenciar a lista com a inclusão e a exclusão de itens a esse vetor. Os principais métodos JavaScript que executam essas tarefas estão indicados na Tabela 6.2.

Tabela 6.2 – Métodos de inclusão e exclusão de itens em vetores

<code>push()</code>	Adiciona um elemento ao final do vetor.
<code>unshift()</code>	Adiciona um elemento ao início do vetor e desloca os elementos existentes uma posição abaixo.
<code>pop()</code>	Remove o último elemento do vetor.
<code>shift()</code>	Remove o primeiro elemento do vetor e desloca os elementos existentes uma posição acima.

TAMANHO DO VETOR: `length`

Uma propriedade importante utilizada na manipulação de vetores é a propriedade `length`, que retorna o número de elementos do vetor. Vamos consultá-la quando quisermos percorrer a lista, realizar exclusões (para verificar antes da exclusão, se a lista está vazia) ou, então, para exibir o número total de itens do vetor. Para percorrer e exibir os elementos do vetor `idades`, podemos utilizar o comando `for`, indicando que a variável de controle `i` começa em 0, e repetir o laço enquanto `i` for menor que `idades.length`.

EXIBIR ITENS DO VETOR

```
const cidades = ["Pelotas", "Porto Alegre", "São Lourenço"]
```

```
for (let i = 0; i < cidades.length; i++) {  
    console.log(cidades[i])  
}
```

```
Pelotas
```

```
Porto Alegre
```

```
São Lourenço
```


toString() e join()

Outra forma de exibir o conteúdo do vetor é pelo uso dos métodos `toString()` e `join()`. Eles convertem o conteúdo do vetor em uma string, sendo que no método `toString()` uma vírgula é inserida entre os elementos e no `join()` podemos indicar qual caractere vai separar os itens.

Métodos JavaScript iniciados por . (ponto)

Vários métodos JavaScript iniciam por ponto e agem sobre a variável associada. Métodos indicam uma ação, já as propriedades (como, length, por exemplo) apenas retornam algo sobre as variáveis ou vetores.

Exemplos de métodos:

- .toFixed(num) => define o número de casas decimais de um número
- .repeat(num) => repete o(s) caractere(s) o número de vezes informado
- .padEnd(num) => acrescenta espaços (ou outro caracter) no final até atingir o número
- .padStart(num) => acrescenta espaços (ou outro caracter) no início até atingir o número